

Esame di Calcolatori Elettronici T

15 Febbraio 2017 (Ing. Informatica)

Esercizio 1

Progettare un sistema basato su un microprocessore DLX a 2 GHz dotato di **768 MB di EPROM** mappata negli indirizzi bassi e **128 MB di RAM** mappata negli indirizzi alti. Nel sistema sono presenti **4 periferiche in input a 8 bit**, già progettate, denominate **INPUT_PORT_i** ($i=[A,B,C,D]$) in grado di comunicare con l'esterno mediante il protocollo di *handshake*.

Sin dall'avvio, ed **esattamente ogni secondo**, dovrà essere valutato (con l'ausilio opportune reti logiche di supporto) quali porte erano disponibili per un trasferimento in quel preciso istante e, con il minimo numero di istruzioni e nel minor tempo possibile, dovranno essere eseguite le letture da tali porte (i.e., quelle effettivamente disponibili) con un unico ciclo di bus. I byte letti dalle porte in input che erano pronte dovranno essere memorizzati, nel minor tempo possibile e con un unico ciclo di bus, agli indirizzi **FFFF1000h** (dato associato alla porta A), **FFFF1001h** (dato associato alla porta B), **FFFF1002h** (dato associato alla porta C) e **FFFF1003h** (dato associato alla porta D). Per ogni porta in input, in caso di lettura non eseguita, non dovrà essere modificato il contenuto della memoria associata alla medesima porta. Inoltre, nei casi in cui non sia possibile leggere da nessuna delle quattro porte: **i)** non dovrà essere interrotto il normale flusso di esecuzione del DLX **ii)** dovrà essere invertito lo stato di un led (inizialmente spento).

Tutte le periferiche saranno utilizzate unicamente per le finalità indicate nel testo.

- **Descrivere sinteticamente la soluzione** che s'intende realizzare indicando **chiaramente quali sono i segnali di chip-select**
- Progettare il sistema **minimizzando le risorse necessarie e risolvendo eventuali criticità**
- Scrivere il codice dell'*interrupt handler*, **commentando in modo chiaro ogni istruzione**, che consente di gestire il problema in accordo alle specifiche assumendo che i registri da R20 a R30 possano essere utilizzati senza la necessità di dover preservare il loro contenuto
- Indicare le espressioni di decodifica e il range di indirizzi di tutte le periferiche, le memorie e i segnali
- Soluzioni **interamente software NON saranno considerate valide**

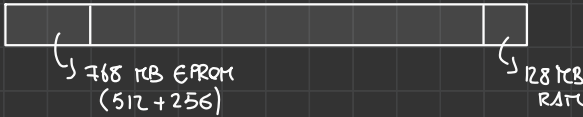
Esercizio 2

- a) Cosa si intende per mapping di memorie a indirizzi non allineati?
- b) Spiegare in dettaglio, con un esempio, come può essere realizzato.

Esercizio 3

- a) Quali soluzioni software esistono per gestire le alee di controllo?
- b) E' necessario impostare l'unità di controllo in modo opportuno?

MAPPING



EPROM - 512 : $0 \times 0000\ 0000 \rightarrow 0 \times 1FFFFFFF$ (4×128)

EPROM - 256 : $0 \times 2000\ 0000 \rightarrow 0 \times 2FFFFFFF$ (4×64)

RAM - 128 : $0 \times F800\ 0000 \rightarrow 0 \times FFFFFFFF$ (4×32)

INPUT_A : $0 \times 8000\ 0000$

INPUT_B : $0 \times 8000\ 0001$

INPUT_C : $0 \times 8000\ 0002$

INPUT_D : $0 \times 8000\ 0003$

RESET_SW : $0 \times 8000\ 0004$

Indice che è
stato eseguito una
lettura della porte
in INPUT

Codifichiamo, a ogni
secondo, quali porte
erano pronte

CS_RAM_0 = BA31 BA30 B6 0 (IN_HANDLER + IN_HANDLER · INT_INPUT_A_READY)

CS_RAM_1 = BA31 BA30 B6 1 (IN_HANDLER + IN_HANDLER · INT_INPUT_B_READY)

CS_RAM_2 = BA31 BA30 B6 2 (IN_HANDLER + IN_HANDLER · INT_INPUT_C_READY)

CS_RAM_3 = BA31 BA30 B6 3 (IN_HANDLER + IN_HANDLER · INT_INPUT_D_READY)

CS_RESET_SW = BA31 · BA30 · BA2 · RESETD

CS_INPUT_PORT = BA31 · BA30 · BA2

Condizioniamo le read
per ridurre le istruzioni
nell'handler

CS_INPUT_A = CS_INPUT_PORT · INT_INPUT_A_READY · BE 0

CS_INPUT_B = CS_INPUT_PORT · INT_INPUT_B_READY · BE 1

CS_INPUT_C = CS_INPUT_PORT · INT_INPUT_C_READY · BE 2

CS_INPUT_D = CS_INPUT_PORT · INT_INPUT_D_READY · BE 3

CS_EPROM_L_0 = BA31 BA29 BE 0

CS_EPROM_L_1 = BA31 BA29 BE 1

CS_EPROM_L_2 = BA31 BA29 BE 2

CS_EPROM_L_3 = BA31 BA29 BE 3

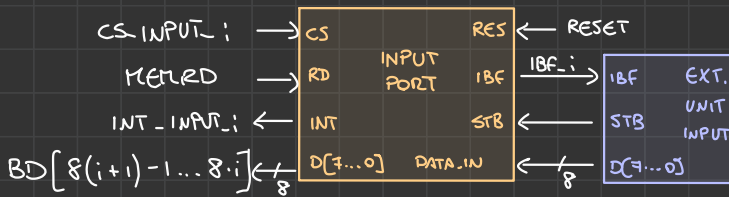
CS_EPROM_H_0 = BA31 BA29 BE 0

CS_EPROM_H_1 = BA31 BA29 BE 1

CS_EPROM_H_2 = BA31 BA29 BE 2

CS_EPROM_H_3 = BA31 BA29 BE 3

PORTE IN INPUT

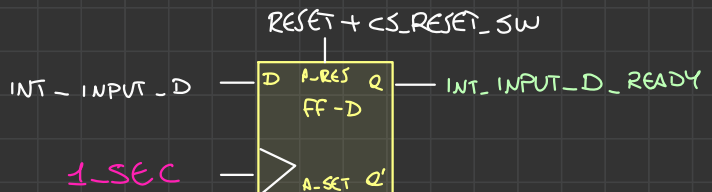
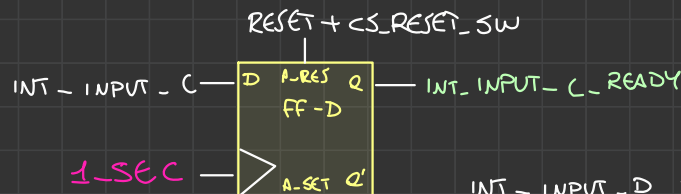
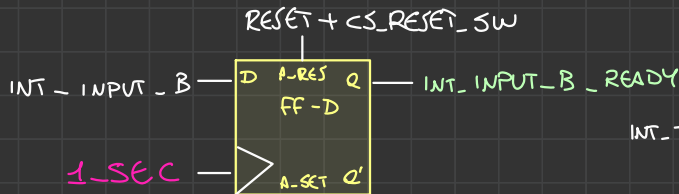
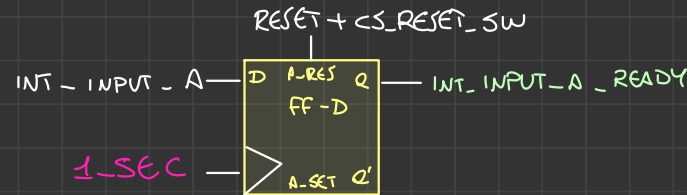


I trasferimenti devono avvenire ogni secondo



↳ Usato per:

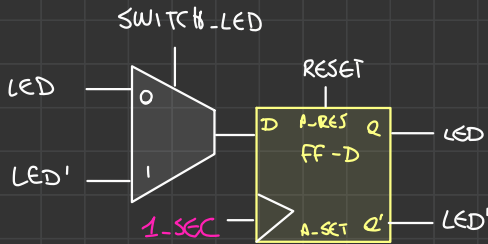
- Campionare i segnali di interrupt generati dalla porte
- Invertire lo stato del led



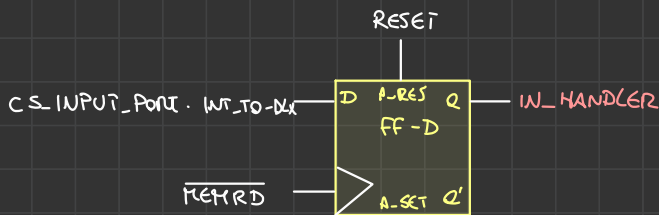
Queste reti: campiano ogni secondo gli interrupt generati dalla quattro porte

$$\text{INT_TO_DLX} = \text{INT_INPUT_A_READY} + \text{INT_INPUT_B_READY} + \text{INT_INPUT_C_READY} + \text{INT_INPUT_D_READY}$$

LED



$$\text{SWITCH_LED} = \text{NOT} (\text{INT_INPUT_A} + \text{INT_INPUT_B} + \text{INT_INPUT_C} + \text{INT_INPUT_D})$$



CODICE

0h	LHI	R21, 0x8000	
4h	LW	R22, 0x0000(R21)	Legge dalle porte abilitate e asserisce IN_HANDLER
8h	LHI	R23, 0xFFFF	
Ch	SW	R22, 0x1000(R23)	Scrive la word a 0xFFFF1000, scrivendo solo sui bus dati dei CS delle RAM abilitate dagli INT_INPUT_i-READY
10h	LW	R25, 0x0004(R21)	Resette in modo asincrono i quattro segnali INT_INPUT_i-READY. IN_HANDLER torna a 0
14h	RFE		